REAL TIME VEHICLE TRACKING SYSTEM WITH DATA LOGGING AND GLASS BREAK DETECTOR

B.Tech in Electronics and Communication Engineering

By

Swapnila Satapathy (111E10260)
Suroshree Das (111ECO208)

Under supervision of

Dr. Santos Kumar Das

Department of Electronics and Communications Engineering

National Institute of Technology, Rourkela

May,2015

Dept. of Electronics & Communication Engineering,
NIT Rourkela

CERTIFICATE

This is to affirm that postulation entitled "Real-time Vehicle Tracking System with Data Logging and Glass Break Detector" has been finished by Swapnila Satapathy, Roll Number 111EI0260 and Suroshree Das , Roll Number 111EC0208, Department of Electronics and Communication Engineering ,National Institute of Technology, Rourkela, India ,amid the period July 2014-April 2015 for the Final Year Project 2014-15 under the supervision of Dr. Santos Kumar Das.

Date:- 10/05/2015

Place:- Rourkela

(Dr. Santos Kumar Das)
Electronics & Communication Engineering
NIT Rourkela

PREFACE

These days Embedded framework has caught the field of industry and in addition residential or useful existence of people. Looking at late improvement in the field or gadgets and PCs, we can simply say that they have caught for all intents and purposes each method of human's life and have given another approach to it. Thinking seriously about, the need to know the whereabouts the specific vehicles or vehicles connected with an unequivocal vision (army, mining, private and so forth) with the assistance of an Embedded System. A Real Time Vehicle Tracking framework with Data logging and Glass Break Detector joined to an Arduino load up is a to a great degree minimal bit of an Electronic instruments set that can all the while send the precise area of the vehicle to the registered versatile or store the areas persistently in a specific site and inform the client on occurance of a mishap or robbery .

Finally we feel very much satisfied in presenting this project, which would be of great use to our society.

ACKNOWLEDGEMENTS

We are greatly thankful to our venture guide, Prof. Santos Kumar Das for his savvy recommendations on the undertaking work and for managing us amid the venture with her support, backing and collaboration. We might want to pass on our sincerest appreciation and obligation to all our faculty members and staffs of Department of Electronics and Communications Engineering, NIT Rourkela, who demonstrated their extraordinary endeavors and direction amid needed times without which it would have been exceptionally hard to do our venture work. In addition, a gathering of this nature could never have been endeavored with our reference to the works of others. We recognize our commitment to every one of them. At last, we might likewise want to expand our heart- felt because of our family for their ethical backing and love.

CONTENTS

ABSTRACT

A low valued following framework connecting with satellites of the Global Positioning System (GPS) is attractive for applications in regards to radiosondes, sonobuoys and so on. The following framework utilizes a sensor bounced on every item which digitally tests the GPS satellite flags and stores them in a buffer. The digital samples are then sent, at a rate lower than the rate at which the GPS satellite signs were examined, over an information telemetry connection, interlarded with other telemetry information. The GPS information is analyzed in a processing workstation where the position and speed of the sensor, at the time the information was inspected, is ascertained. The buffer in the sensor is refreshed, and the workstation intermittently figures the new position and speed of the sensor. Differential rectifications are additionally done at the workstation to help in data acquisition and to build the exactness of the position and velocity. Cutting edge vehicle following framework uses GPS (Global Positioning System) and GSM (Global System for Mobiles) innovation for finding the vehicle and give security to the vehicle. A glass break detector is a sensor or transducer which is utilized as a part of electronic thief cautions that watches if a sheet of glass is smashed or disturbed or broken. These sensors are largely utilized in close glass entryways or in cars to identify if a stealer broke the glass and entered or hurt the auto or a mishap happened.

RUN DOWN OF FIGURES

CHAPTER 1- INTRODUCTION

1.1 INTRODUCTION TO VEHICLE TRACKER AND GLASS BREAK DETECTOR

1.1.1 What is a Vehicle tracker?

The current location of the vehicle is determined using GPS (Global Positioning System) and the location co-ordinates are transmitted using different technologies like SMS(Short Message Service)using GSM , GPRS(General Packet Radio Service). The vehicle location is useful in successful tracking of the vehicle at times of theft and accidents. It has its application in tracking mobile assets, costly construction equipments. We can also use it for managing large driver and crewing staff in travel and transport business.

1.1.2 What is a glass break detector?

A glass break detector for detecting the breaking of car glass door uses an acoustic transducer to get a wide band frequency response and a dual channel filter circuitry and a signal processing unit along with it. The low frequency channel detects the low inward compression and the high frequency detects the other characteristics of a glass breaking. With the help of a timing circuit the low frequency flex is detected with the high frequency characteristics shortly coming after. After the timing condition satisfies which is done here with the help of an Arduino-uno ,the alarm is triggered.

Main objective of this detector is that it senses the frequency characteristics of the glass breaking and provides an alarm upon satisfaction of the two frequency criterion within the predefined time gap given in the Arduino Uno. The current experiment results from the discovery that breaking glass produces highly characteristic patterns of acoustic waves, and

particularly , produces a positive low frequency acoustic wave and high frequency acoustic waves that follow the initial low frequency phenomena

For some time invasion detectors have made use of the occurrence that the opening of a door produces an infrasonic pressure wave that can be detected by a microphone or an acoustic transducer that has a frequency response in the region of one to five or ten cycles. An example of this is shown in U.S. Pat. No. 4,853,677. The Yarbrough device (U.S. Pat. No. 4,853,677) uses a glassbreak detector circuitry that is coupled to the microphone. A high frequency or a low frequency event will trigger the alarm if either of them produces the desired frequency spectrum set in the filters. Again , it has been identified that the opening of a door produces negative air pressure in the first and sound detectors which are invasion detectors have been designed to make use of this fact. An example is shown in U.S. Pat. No.4,991,145.

1.1.3 Applications of Vehicle Tracker:

- **Recovering Stolen Vehicle** : Both consumer and commercial vehicles can be attached to GPS with cellular transmitter to access police to do tracking and recovery.
- **Trailer Tracking**: Haulage and Logistics companies essentially operate trucks with detachable load carrying units. There are different types of trailer used for applications like flat bed, refrigerated and curtain sider.
- Instantaneous location target with an accuracy of 2 to 3 meters helps in providing emergency customer services i.e. location of the ruined vehicle can be used to get ambulances to provide emergency aid.
- **Monitoring vehicles online** : This helps in navigation.

- **Field Service Management**: Companies with services such as repair or maintenance prefer to plan workers' time and schedule for efficient subsequent visits. Vehicle tracking permits companies to locate easily and quickly an engineer and send the closest one to meet a new customer or provide site information.
- **Data logging** into a website or another server for tracking the movement of vehicles.
- **Monitor the movement of vehicles**: This helps in fleet management by decreasing various costs while delivering assets safely and quickly. The real time reports from the tracker assists in better planning and estimating and therefore increasing customer satisfaction and trust with the company.

1.1.4 Applications of glass break detector:

- Avoid undesired theft and burglary.
- Getting notification of an accident instantaneously.

1.2 OBJECTIVE

In this project we aim in developing a product for real-time vehicle tracking system with data logging and glass break detector. Vehicle tracking provides real time location coordinates of its position, this co-ordinates are transmitted using a GSM/GPRS TTL SIM900 modem to a server which logs the location in a tabular format with current time stamp.

The glass break detector detects high frequency noise signal corresponding to a glass breakage. This can be helpful in detection of accidents. On detection of a glass break, the current position co-ordinates are transmitted via SMS to any given mobile user, so as to provide immediate aid.

1.3 SCOPE OF PROJECT

The project is concerned with the designing of a real-time tracking and data logging system with a glass break detector. It will cover the scopes as follows:

- Developing a code in ARDUINO1.5.5 for tracking the exact location latitude and longitude using GPS UART receiver.
- Code for transmitting the data through SMS using GSM/GPRS TTL SIM 900A modem.
- Data-logging by interfacing GPRS Modem with 2G connection and transmiiting data to PushingBox server.
- Updating Google Form with location co-ordinates sent by PushingBox.

1.4 PROBLEM STATEMENT

Classical vehicle tracking system are used by cars in order to track its location. This location in the form of geographical co-ordinates are stored in a tabular format, which could be used for tracking the current location as well as monitor its movement from one location to another. For data logging it depends on GPRS connection which is easily available at cheap tariff rates by the different service providers. Along with this we have also developed the software model of the glass break detector which acts as an alarm during accidents, sending the current location of the vehicle via SMS to a given user/users for immediate aid. Vehicle trackers available in market are costly and perform inefficiently at places. They require periodic maintenance and are prone to malfunctioning.

CHAPTER-2
HARDWARE DEVELOPMENT AND SIMULATIONS

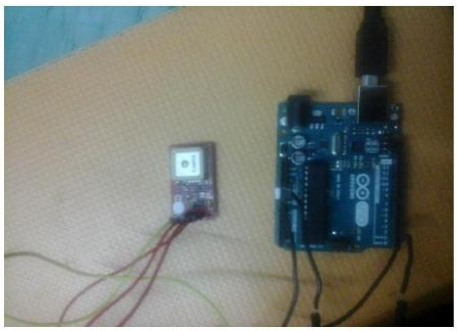
2.1 INTRODUCTION

Here we give a brief overview of the different hardware components used along with their specifications. Then we show the interfacing of the different components to build the final product. The hardware components used are

- GPS UART Modem(G.top010)
- SIM900 TTL(GPRS-GSM) Modem
- Arduino UNO –R3

2.2 GPS UART Modem

(Figure2.1: GPS MODEM)



(Figure 2.2: GPS joined with Arduino)

This is a high gain GPS Receiver with 4-pin 2.5mm pitch strip. Receiver is made with third generation POT GPS module. The built in 3V - 5V level convertor enables to interface with 5V controllers. Its low pin count and bread-board friendly features make it easy to interface . The 4

Pins are 5V, TX, RX, GND, no internal setting is required to arrange, just by plugging into the power (5V), the data (NMEA 0183) will be ready to be transmitted at TX pin.

2.2.1 FEATURES

- 5V DC supply (60 mA max current).
- TTL asynchronous interfacing.
- Output Baud rate: 9600 bps .
- Output format-NMEA0183 .
- Standard 4-pin interface (2.54mm).
- Provided with two type strips.
- Focussing Media-Tek Chip Architecture.
- Patch Antenna (dimensions-25mm x 25mm x 4mm).
- Low Power Consumption.
- L1 Frequency, C/A code, 51-channels.
- Better Sensitivity: better urban performances.
- Accuracy: < 3m CEP (50%) without SA

2.2.2 OUTPUT OF GPS MODULE

$GPVTG,183.80,T,,M,0.36,N,0.67,K,A*3B.649,V,,,,,0.00,0.00,080180,,,N*4F

$GPGGA,113329.000,2215.0789,N,08454.1140,E,1,5,4.32,249.9,M,-59.3,M,,*78,K,N*32

$GPGGA,112948.649,,,,,0,0,,,M

$GPGSA,A,3,19,08,03,27,07,,,,,,,,4.43,4.32,0.98*04A,A,1,,,,,,,,,,,,,*1E

$GPVTG,0.00,T,,M,0.

$GPGSV,3,1,10,07,51,352,38,19,50,051,39,40,47,236,29,03,34,040,38*75,,,,,0,0,,,M,,M,,*4D

$GPGSA,A,1

$GPGSV,3,2,10,08,28,325,27,27,17,039,37,16,01,057,18,28,,,19*48,V,,,,,0.00,0.00,080180,,,N*49

$GPGSV,3,3,10,09,,,21,10,,,16*74.00,N,0.00,K,N*32

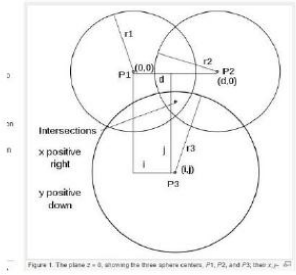
$GPRMC,113330.000,A,2215.0787,N,08454.1139,E,0.10,184.28,080414,,,A*69

2.2.3 Working of GPS Modem

- A GPS receiver's main objective is to locate four of more satellites among the 27 satellites orbiting the earth to correctly calculate the exact location of an object.
- It figures out the distance of each satellite from itself and uses this information to deduce its own location. This mathematical operation is based on a principle called Trilateration.
- At a particular time the satellite begins transmission of a digital pattern which is known as pseudo-random code. On reaching the receiver the pattern lags in time due to travel delay and the length of the delay is equal to the travel time of the signal. The receiver multiplies this time with the speed of light to determine the approximate distance travelled by the signal and thus calculates its position.

2.2.4 TRILATERATION

- It is the mathematical model used to measure the absolute or relative locations of points by measuring the distances from a particular point using the geometry of circles.



(Figure 2.3 : Trilateration)

- In 3-D geometry if the point lies on the surface of 3 spheres, then using their centers and their radii, the correct location of the GPS receiver can be calculated.

$$r1 = x^2 + y^2 + z^2 ; \quad r2 = (x - d)^2 + y^2 + z^2$$

$$r3 = (x - i)^2 + (y - j)^2 + z^2$$

$$r1^2 - r2^2 + d^2 = 2\text{xd}$$

On back substituting this value

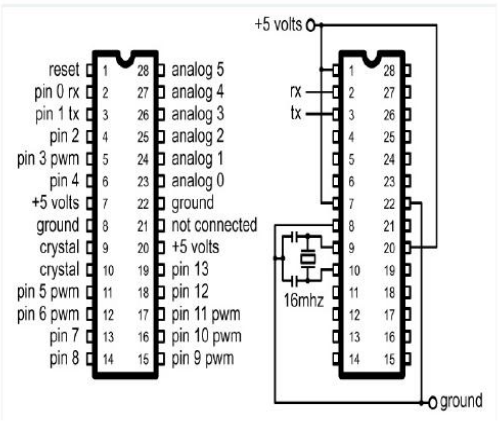
$$x = \frac{r1^2 - r2^2 + d^2}{2d} \quad ;$$

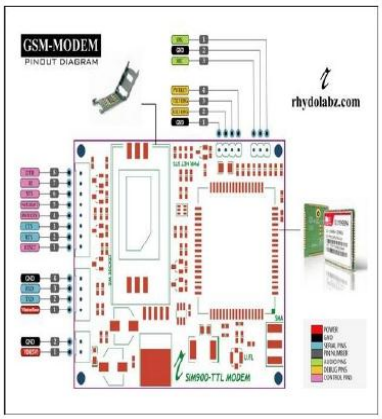
$$y = \frac{r1^2 - r3^2 + i^2 + j^2}{2j} - \frac{\text{ix}}{j}$$

2.2.5 NMEA STRINGS

NMEA strings stands for National Marine Electronics Association, a group for Electronics Trade that implements standards and rules. NMEA basically sends a line of data that is totally self contained and unique from other sentences. Each line starts with a '$' and ends with line feed and contains no more than 80 characters. The location co-ordinates received by the GPS Modem is in the form of NMEA Strings. Here we have used '$GPRMC' string to find the position coordinates.

$GPRMC,113330.000,A,2215.0787,N,08454.1139,E,0.10,184.28,080414,,,A*69





(Figure 2.5 :Arduino Pin Diagram)

2.4 ARDUINO UNO R3

2.4.1 OVERVIEW

The Arduino Uno is a microcontroller board based on the ATmega328 . It has 14 digital input/output pins (6 can be used for Pulse Width Modulation outputs), 6 analog inputs, 16 MHz resonator, a Universal Serial Bus port, power jack, ICSP header, and one reset button. The ATmega328 has 32 KB (among which 0.5 KB is used for the boot loader). It has 2 KB of Static-RAM and 1 KB of EEPROM .

2.4.2 POWER

The Arduino Uno can be supplied power with the help of the Universal Serial Bus connection or an external power supply of 5 volts. External power source may be an AC-to-DC adapter or a

battery. The adapter is connected by connecting a 2.1mm plug into the power jack. Leads of a battery can be used as the Ground and Vin(supply) pin heads of the power connector.

2.4.3 COMMUNICATION

It has many facilities for its communication with a computer, another Arduino board, or microcontrollers. The ATmega328 depicts UART- TTL serial communication, which is available on pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels the serial communication over USB and behaves as a virtual com port to the software. The '16U2 firmware provides the standard USB drivers. The Arduino usess the serial monitor that allows text data to be sent through . The RX and TX LEDs flashes when data is transmitted via the USB to serial chip and the computer.

2.4.4 PHYSICAL CHARACTERISTICS

The length and width of the Arduino Uno PCB are 2.8 and 2.2 inches respectively, excluding the USB connector and power jack. Four screw holes are there on the board to attach to a surface . The distance between the digital pins 7 and 8 is around 160 mil (0.16") which is not an even multiple of the 100 mil i.e. the spacing of the other pins.

2.5 GPS, GPRS WITH ARDUINO INTERFACING

2.5.1 INTRODUCTION

Finally we interface all the above mentioned hardware in order to develop the final product. Each of the GPS and the GPRS codes were combined to give the final result. Basically the GPS provides the location co-ordinates which are then stored in a google form in tabular format.

The transmitting of the data to the google form is done by the GPRS modem. Here we have implemented a intermediated application provider 'PushingBox.com'.

2.5.2 PUSHING BOX

PusingBox ia a cloud based service used for sending notifications using APi calls. Triggered using a HTTP request (Get/Post) it can send seversl requests simultaneously. We send out data via GPRS to PushingBox. In PushingBox the directions for storing data in the google form are given, hence it acts as an intermediator. First generate a google form("Arduino Data") with three columns in our drive. The unique form key generated along with the column names generated by HTML was noted down. In PushingBox we generate a service ("Arduino Data") using the form key of our google form. Then we generate a scenario and give the column name as its input. On completion of the scenario a unique device id is generated. Finally the scenario is linked with the service. Then we go for the final coding.

Here we have generated two scenarios with corresponding two services for transmitting the latitude and longitude. Here the values are transmitted one after the other to the pushing box server from where it is stored in the google form.

2.5.3 ARDUINO CODE

```
  #include <SoftwareSerial.h>
SoftwareSerial gprsSerial(7, 8);
int Gpsdata;            // for incoming serial data
 unsigned int finish =0;  // indicate end of message
 unsigned int pos_cnt=0;  // position counter
 unsigned int lat_cnt=0;  // latitude data counter
 unsigned int log_cnt=0;  // longitude data counter
 unsigned int flg   =0;  // GPS flag
 unsigned int com_cnt=0;  // comma counter
 char lat[20];           // latitude array
 char lg[20];

 void Receive_GPS_Data();

void setup()
{

 gprsSerial.begin(19200);
 Serial.begin(19200);

 Serial.println("Config SIM900...");
 delay(2000);
 Serial.println("Done!...");
 gprsSerial.flush();
 Serial.flush();
 gprsSerial.println("AT+CPIN?");
 delay(100);
 toSerial();
 gprsSerial.println("AT+CREG?");
 delay(100);
 toSerial();
```

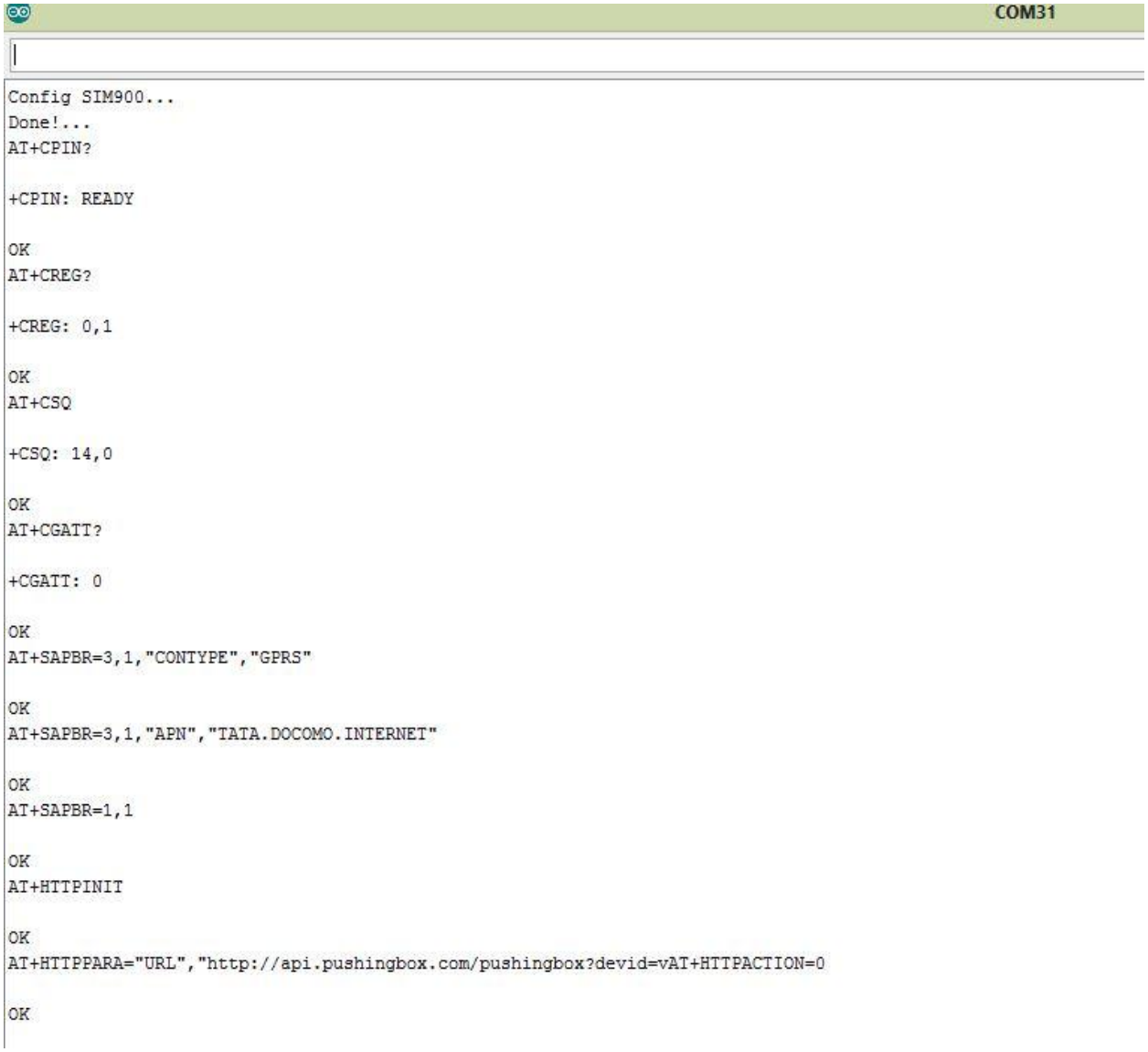
```
 gprsSerial.println("AT+CSQ");
 delay(100);
 toSerial();

 gprsSerial.println("AT+CGATT?");
 delay(100);
 toSerial();

 gprsSerial.println("AT+SAPBR=3,1,\"CONTYPE\",\"GPRS\"");
 delay(2000);
 toSerial();


 gprsSerial.println("AT+SAPBR=3,1,\"APN\",\"TATA.DOCOMO.INTERNET\"");
 delay(2000);
 toSerial();


 gprsSerial.println("AT+SAPBR=1,1");
 delay(2000);
 toSerial();
}


void loop()
{
  Receive_GPS_Data();
     String lat1(lat);
  Serial.println(lat1);

    finish = 0;pos_cnt = 0;
```

```
  gprsSerial.println("AT+HTTPINIT");
  delay(2000);
  toSerial();

  gprsSerial.println("
AT+HTTPPARA=\"URL\",\"http://api.pushingbox.com/pushingbox?devid=vCD8944C5089CEFC&statu
s=" + lat1 + "\" ");
  delay(2000);
  toSerial();

  gprsSerial.println("AT+HTTPACTION=0");
  delay(6000);
  toSerial();

  gprsSerial.println("AT+HTTPREAD");
  delay(1000);
  toSerial();

  gprsSerial.println("");
  gprsSerial.println("AT+HTTPTERM");
  toSerial();
  delay(300);

  gprsSerial.println("");
  delay(10000);
}

void toSerial()
{
  while(gprsSerial.available()!=0)
  {
```

```
  Serial.write(gprsSerial.read());
 }
}

void Receive_GPS_Data()
 {
  while(finish==0){
   while(Serial.available()>0){        // Check GPS data
    Gpsdata = Serial.read();
    flg = 1;
   if( Gpsdata=='$' && pos_cnt == 0)   // finding GPRMC header
     pos_cnt=1;
   if( Gpsdata=='G' && pos_cnt == 1)
     pos_cnt=2;
   if( Gpsdata=='P' && pos_cnt == 2)
     pos_cnt=3;
   if( Gpsdata=='R' && pos_cnt == 3)
     pos_cnt=4;
   if( Gpsdata=='M' && pos_cnt == 4)
     pos_cnt=5;
   if( Gpsdata=='C' && pos_cnt==5 )
     pos_cnt=6;
   if(pos_cnt==6 &&  Gpsdata ==','){   // count commas in message
     com_cnt++;
     flg=0;


   if(com_cnt==3 && flg==1){
    lat[lat_cnt++] =  Gpsdata;        // latitude
    flg=0;
   }
```

```
    if(com_cnt==5 && flg==1){
      lg[log_cnt++] =  Gpsdata;        // Longitude

      flg=0;
    }


    if( Gpsdata == '*' && com_cnt >= 5){
      com_cnt = 0;                    // end of GPRMC message
      lat_cnt = 0;
      log_cnt = 0;
      flg    = 0;
      finish  = 1;
      delay(5000);
    }

   }
 }
}
 }
```

2.5.4: OUTPUTS


COM31
Config SIM900...
Done!...
AT+CPIN?
+CPIN: READY
OK
AT+CREG?
+CREG: 0,1
OK
AT+CSQ
+CSQ: 14,0
OK
AT+CGATT?
+CGATT: 0
OK
AT+SAPBR=3,1,"CONTYPE","GPRS"
OK
AT+SAPBR=3,1,"APN","TATA.DOCOMO.INTERNET"
OK
AT+SAPBR=1,1
OK
AT+HTTPINIT
OK
AT+HTTPPARA="URL","http://api.pushingbox.com/pushingbox?devid=vAT+HTTPACTION=0
OK

COM31

```
OK
AT+HTTPPARA="URL","http://api.pushingbox.com/pushingbox?devid=vAT+HTTPACTION=0

OK

+HTTPACTION:0,200,0
AT+HTTPREAD

OK

AT+HTTPTERM

OK

AT+HTTPINIT

OK
AT+HTTPPARA="URL","http://api.pushingbox.com/pushingbox?devid=vAT+HTTPACTION=0

OK

+HTTPACTION:0,200,0
AT+HTTPREAD

OK

AT+HTTPTERM

OK

AT+HTTPINIT

OK
AT+HTTPPARA="URL","http://api.pushingbox.com/pushingbox?devid=vAT+HTTPACTION=0

OK

+HTTPACTION:0,200,0
```

Autoscroll

CHAPTER-3
SOFTWARE DEVELOPMENT AND SIMULATIONS

3.1 INTERFACING GPRS AND GSM MODEM WITH ARDUINO

```
#include <SoftwareSerial.h>
 // String lat="name";
#define Connect "AT+HTTPPARA=\"URL\",\"http://api.pushingbox.com/pushingbox?devid=vD3470D1FE5B4772&status=" + lat1 + "\" "
SoftwareSerial gprsSerial(7, 8);
int Gpsdata;
 unsigned int finish =0;
 unsigned int pos_cnt=0;
 unsigned int lat_cnt=0;
 unsigned int log_cnt=0;
 unsigned int flg   =0;  /
 unsigned int com_cnt=0;
 char lat[20];
 char lg[20];
//String lat1(lat);
// String lg1(lg);
 void Receive_GPS_Data();

void setup()
{

 gprsSerial.begin(19200);
 Serial.begin(19200);

 Serial.println("Config SIM900...");
 delay(2000);
 Serial.println("Done!...");
```

```
gprsSerial.flush();
Serial.flush();
gprsSerial.println("AT+CPIN?");
delay(100);
toSerial();
gprsSerial.println("AT+CREG?");
delay(100);
toSerial();
 gprsSerial.println("AT+CSQ");
delay(100);
toSerial();


gprsSerial.println("AT+CGATT?");
delay(100);
toSerial();



gprsSerial.println("AT+SAPBR=3,1,\"CONTYPE\",\"GPRS\"");
delay(2000);
toSerial();

// bearer settings
gprsSerial.println("AT+SAPBR=3,1,\"APN\",\"TATA.DOCOMO.INTERNET\"");
delay(2000);
toSerial();

// bearer settings
```

```
 gprsSerial.println("AT+SAPBR=1,1");
 delay(2000);
 toSerial();
}


void loop()
{

    // Serial.print("Longitude : ");
    // Serial.println(lg1);
  Receive_GPS_Data();
     String lat1(lat);
  Serial.println(lat1);
   //  Serial.print("Latitude : ");


    finish = 0;pos_cnt = 0;
  // initialize http service
  gprsSerial.println("AT+HTTPINIT");
  delay(2000);
  toSerial();


  // set http param value
 //
gprsSerial.println("AT+HTTPPARA=\"URL\",\"http://api.pushingbox.com/pushingbox?devi
d=vD3470D1FE5B4772&status=%s\"");
  gprsSerial.println(Connect);
  gprsSerial.println("
AT+HTTPPARA=\"URL\",\"http://api.pushingbox.com/pushingbox?devid=vCD8944C5089
CEFC&status=" + lat1 + "\" ");
  delay(2000);
```

```
  toSerial();

  // set http action type 0 = GET, 1 = POST, 2 = HEAD
  gprsSerial.println("AT+HTTPACTION=0");
  delay(6000);
  toSerial();

  // read server response
  gprsSerial.println("AT+HTTPREAD");
  delay(1000);
  toSerial();

  gprsSerial.println("");
  gprsSerial.println("AT+HTTPTERM");
  toSerial();
  delay(300);

  gprsSerial.println("");
  delay(10000);
}

void toSerial()
{
 while(gprsSerial.available()!=0)
 {
  Serial.write(gprsSerial.read());
 }
}
```

```
void Receive_GPS_Data()
 {
  while(finish==0){
    while(Serial.available()>0){         // Check GPS data
     Gpsdata = Serial.read();
     flg = 1;
    if( Gpsdata=='$' && pos_cnt == 0)   // finding GPRMC header
      pos_cnt=1;
    if( Gpsdata=='G' && pos_cnt == 1)
      pos_cnt=2;
    if( Gpsdata=='P' && pos_cnt == 2)
      pos_cnt=3;
    if( Gpsdata=='R' && pos_cnt == 3)
      pos_cnt=4;
    if( Gpsdata=='M' && pos_cnt == 4)
      pos_cnt=5;
    if( Gpsdata=='C' && pos_cnt==5 )
      pos_cnt=6;
    if(pos_cnt==6 &&  Gpsdata ==','){   // count commas in message
      com_cnt++;
      flg=0;


    if(com_cnt==3 && flg==1){
     lat[lat_cnt++] =  Gpsdata;         // latitude
     flg=0;
    }
   // lat[lat_cnt++] ='&';
```

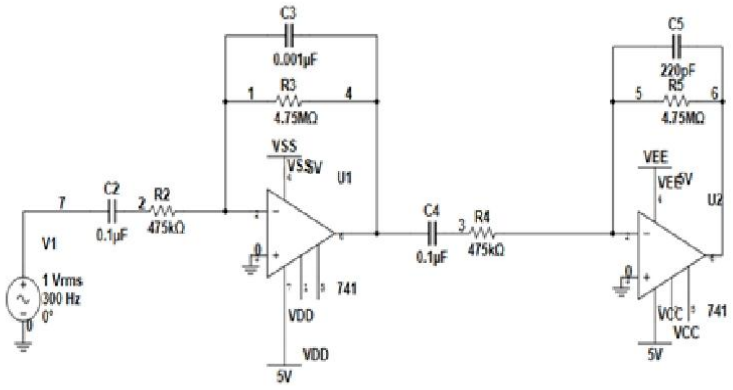
```
    if(com_cnt==5 && flg==1){
      lg[log_cnt++] =  Gpsdata;        // Longitude
    //   lat[lat_cnt++] =Gpsdata;
      flg=0;
     }


     if( Gpsdata == '*' && com_cnt >= 5){
       com_cnt = 0;                    // end of GPRMC message
       lat_cnt = 0;
       log_cnt = 0;
       flg     = 0;
       finish  = 1;
       delay(5000);
     }

   }
 }
}
}
```

3.2 GLASS BREAK DETECTOR

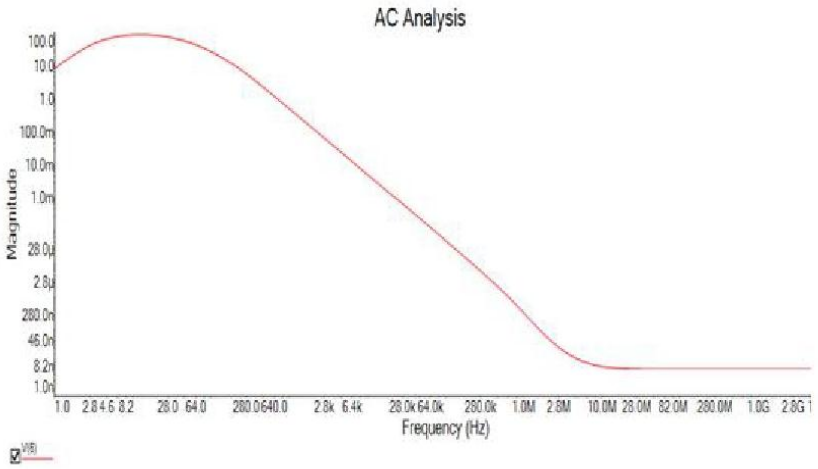
According to the current project, an invasion detector for detecting the breaking of a glass pane uses an acoustic transducer such as a microphone with excellent dynamic characteristics and a signal processing circuit antiphonal to the acoustic transducer for observing a first low frequency negative sound wave which is produced by an inbound flex of the glass and an alarm antiphonal to the signal processing circuit. This system includes a high frequency band-pass filter for detection of high frequency characteristic waves featuring the breaking of glass and a concurrence logic circuit that modifies the alarm when the high frequency sound wave is seen within the preselected time window that begins with a low frequency consequence generated by the breaking of the glass. The alarm can be triggered by testing the high frequency output of the microphone transducer at the time after the initial time window. The logic of this system takes the fact that the characteristic high frequency spectrum of sound waves will follow the negative low frequency wave generated by the inbound flex of the glass pane. A circuit can be provided to enable the alarm upon the detection of negative-going low frequency developments abided by high frequency sounds that will partially enable the alarm. This alarm enabling feature significantly decreases the chance of occurrence of false alarms such as those alarms that would be caused by a opening of a door following, high frequency sounds for example the jangling of keys. The project also capitalizes the fact that, irrespective of the type of glass, the low frequency component of this lies in the frequency region of 50 Hz and 100 Hz and the breaking of glass is always started by a negative compression wave. Infrasonic detectors of the anterior art frequently engaged on the principle that a glass break generates a low frequency sound that vibrates the room, coupling it to the outside world through the bumped door. The problem in this is that such

low frequency vibration can also take place for a large number of occasions which are not related with breaking of the glass.

3.2.1 L F B P Filter



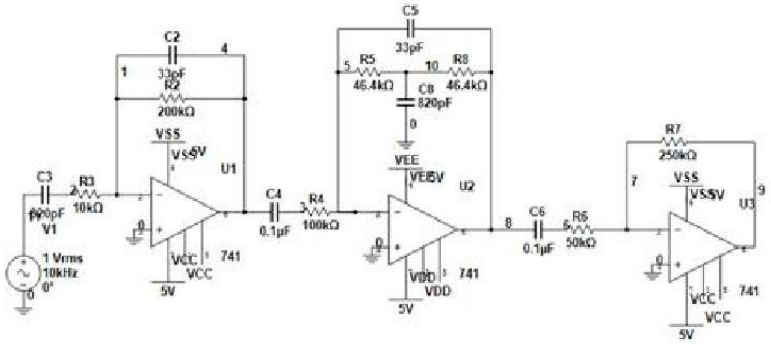
(Figure:3.1: LFBP Filter in Multisim)



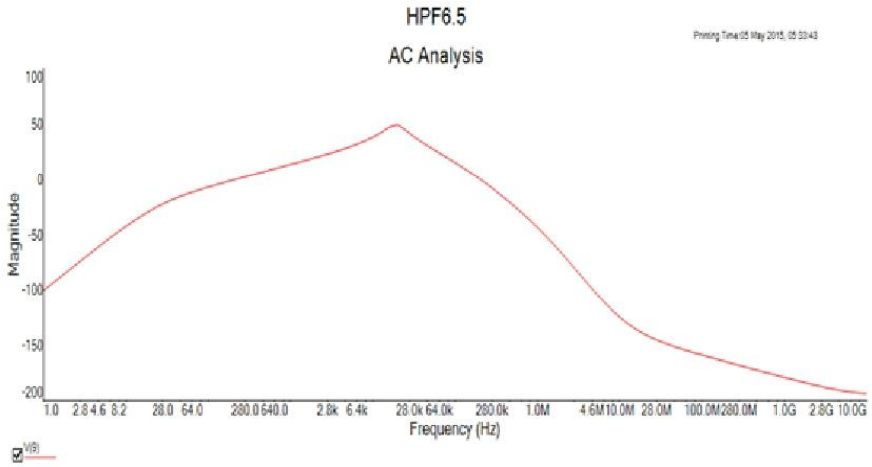
(Figure 3.2: AC characteristics of LFBP Filter in NI Multisim)

This Low Frequency Band Pass Filter is magnifying the frequency range 50 Hz-100Hz. In the first OPAMP set the high pass filter is producing the high pass pole at nearly 3.4 Hz and the low pass filter set is producing the low pass pole at 34 Hz. Then the second OPAMP 's high pass filter is giving a high pass cut off at 3.4 Hz and low pass filter is producing the cutoff frequency at 154 Hz. So this Low Frequency Band pass Filter is of 4 th order and giving output with 6 db gain at some places and 12 db gain at some places.

3.2.2 H F B P Filter



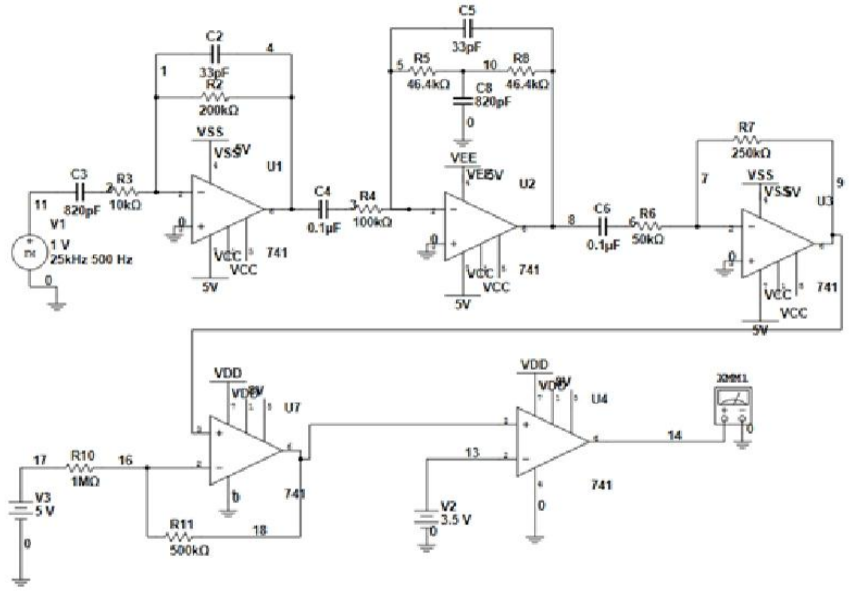
(Figure:3.3: HFBP Filter)



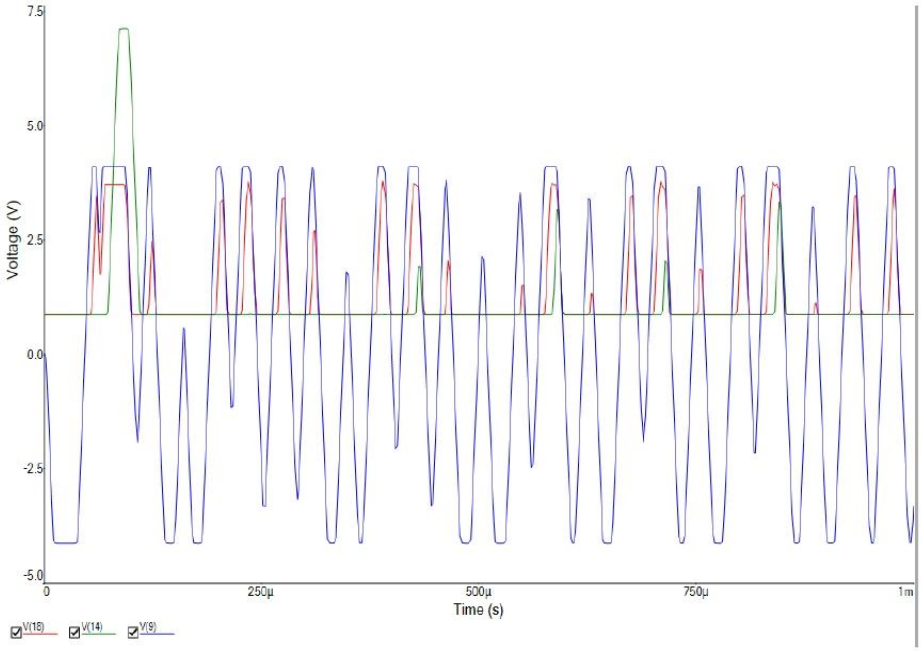
(**Figure 3.4: AC Characteristics of HFBP Filter)**

Here in the High Frequency Band pass Filter we have the first OPAMP's high pass filter 's high pass pole at 19 KHz and the low pass filter 's low pass pole at 24 KHz. The second OPAMP's low pass filter's cutoff frequency is 16 Hz and high pass filter is having its cutoff at 24 Khz . this arrangement is giving minimum of 6 db gain and at 20 Khz the peak is formed.

3.2.3 H F B P Filter with Comparator



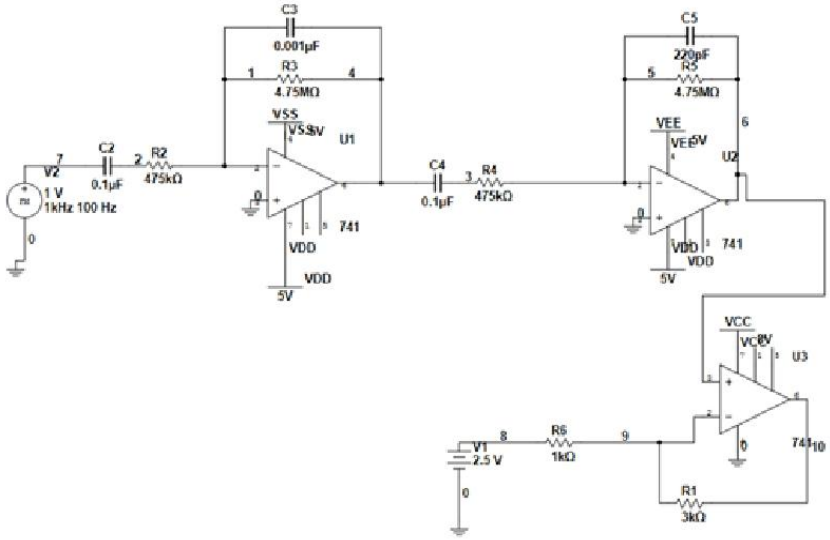
(Figure 3.5 :HFBP Filter with Comparator)



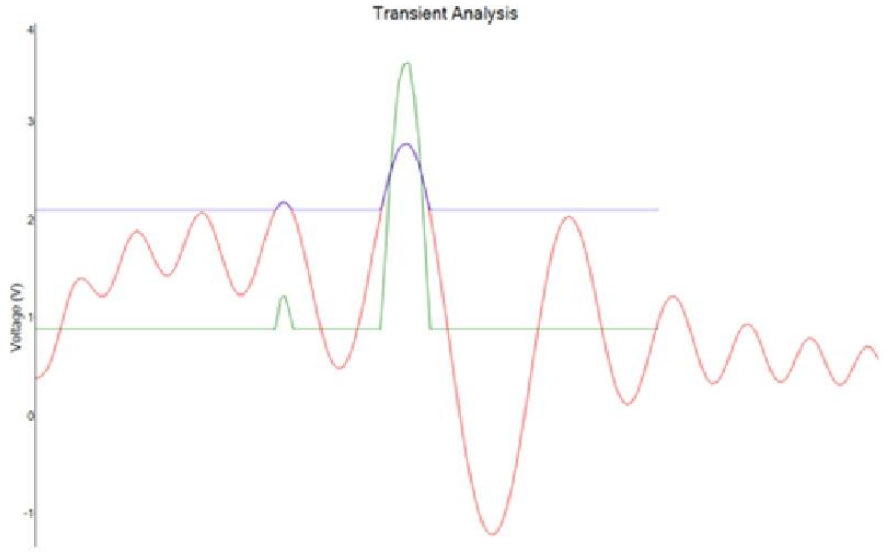
(Figure 3.6:Comparator output of HFBP Filter)

The output of the filter varies from 2.5 volts to 5 volts. We will need various range of volatage depending upon the glass type. Some glass may be giving the output in the 2-4 volts and some others can give more than 4 volts . So depending upon the glass we can change the reference voltage of the second comparator and get only desired output voltage range and amplified version of those signals.

3.2.4 L F BP Filter with Comparator

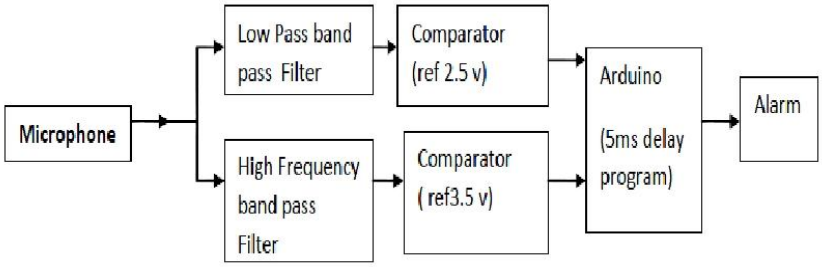


(Figure 3.7:LFBP Filter with Comparator)



(Figure3.8: Comparator output from LFBP Filter)

3.2.5 : Overall Connection Of Detector with Arduino



(Figure 3.9:Overall Block Diagram of Detector)

The low frequency channel detects the low inward compression and the high frequency detects the other characteristics of a glass breaking. With the help of a timing circuit the low frequency flex is detected with the high frequency characteristics shortly coming after. After the timing condition satisfies which is done here with the help of an Arduino-uno ,the alarm is triggered.

CONCLUSION

After proper implementation of GPS,GSM-GPRS and Pushing Box as well NI-MULTISIM Software, we have a total set up which can be used to get the proper locations and store it in google spreadshits and a simulated glass break detector which embeds all our desired characterstics as well as is portable and efficient to serve as the Real Time Vehicle Tracking System and can be used in passing on Alert Messages about the occurrence of a theft to the Registered Mobile Numbers as well as the total travel history of that vehicle can be stored on desired website.

Further testing on the Real Time Vehicle Tracking System with Data logging and Glass Break Detector may show us areas where we need to improve and customize so as to make the above said better and efficient.

REFERENCES

[1] Richard A. Smith , Christopher A. Bernahardt "Dual channel glass break detector":*United State Patent*, Patent no:5,192,931, Date of Patent: Mar.9,1993.

[2] Frank B. Clark, Kenneth K. Lewis; "Glass Break Detector and a Method therefor" : *United States Patent*, Patent no: 5543783, Date of Patent: August 6,1996.

[3] Dennis Ceccic, Hartwell Fong; "Glass Break Sensor" : *United States Patent*, Patent no: 5917410, Date of Patent Jun 29,1999.

[4] "Arobust Glass Break Detector Reference Design", http://www.ti.com/tool/TIDM-GBD-ROBUST .

[5] "Connecting a Parallax GPS module Arduino and Hardware Connections", http://playground.arduino.cc/Tutorials/GPS

[6] "GSM Test GPRS", http://www.arduino.cc/en/Tutorial/GSMToolsTestGPRS

[7] "Glass Break Detector", http://www.best-microcontroller-projects.com/glass-break-detector.html